UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE INFORMÁTICA

SMART CITY BUS: FLEXIBLETRANSPORT.AI FRONT END

Gonçalo Alexandre Moleira Domingos

Mestrado em Engenharia Informática
Especialização em Engenharia de Software

Versão Pública

Trabalho de Projeto orientado por:
Ana Paula Pereira Afonso

2019

Acknowledgments

First and foremost, I must thank my family, specially my parents, my sister and my uncle for the constant support and motivation that allowed me to successfully reach this stage of my life and helped me become who I am today.

I would also like to express my sincere gratitude to Professor Doctor Ana Paula Pereira Afonso for her availability and guidance during this last year and whose help was beyond valuable for me to finish this project. Similarly, I must thank the University of Lisbon for giving me a chance to learn so much about what I like and want to do for the rest of my life.

I am also grateful to the institution where this project was developed, Card4B - Systems, for allowing me to do so, to João Almeida and Jorge Mendes for assisting me in the development of this project and to everyone else inside the institution who made me feel so welcome right from the start and taught me what a good working environment should be like.

Last but not least, I must thank my friends who have contributed so significantly to make me who I am today and who have helped me and supported me in more ways than they could ever possibly imagine.

Thank you.

To my parents and sister

To my grandparents and uncle

Resumo

No âmbito do projeto SMART CITY BUS: FLEXIBLE TRANSPORT.AI FRONT-END desenvolvido na empresa Card4B - Systems, S.A. foi realizado o presente documento para relatar os resultados do referido projeto no âmbito da cadeira de Dissertação/ Projeto de Engenharia Informática do 2º ano do Mestrado em Engenharia Informática, especialização em Design de Software, na Faculdade de Ciências da Universidade de Lisboa. O projeto está diretamente relacionado com a gestão de serviços de transportes públicos e com o melhoramento e aumento da flexibilidade dos mesmos através de mecanismos de recomendação de novas rotas e serviços, focando na componente *front-end* (Android) do sistema, tanto do lado do cliente/passageiro como do lado do motorista.

A dependência dos transportes públicos é indiscutível, tal como a divergência conceitual entre os dois principais tipos de serviços que são oferecidos. O conceito de autocarro ou comboio - serviço regularizado, com percursos e horários previamente definidos - é completamente o oposto de serviços como os taxis ou a UBER, por exemplo - *on-demand*. Qualquer vestígio adicional de flexibilidade acresce no custo do serviço, o que acaba por limitar a sua utilização pelas massas.

Com a evolução das tecnologias e comunicações começa a ser possível ter noção dos traços gerais de movimento e focos de cidadãos na malha urbana. Como tal, é possível dinamizar a gestão dos serviços de transporte coletivo de modo a torná-los mais flexíveis e adaptáveis às necessidades e padrões de mobilidade da população. Não só para a população local, que exige um melhoramento significativo do sistema que já existe e dos serviços que usam no dia-a-dia, mas também o enorme crescimento de turistas observado nos últimos anos implica que deve haver uma reestruturação do sistema de gestão destes serviços de mobilidade, principalmente em pontos chave da cidade.

O objetivo é desenvolver um sistema que será integrado numa plataforma de gestão de serviços de mobilidade urbana que possa detectar e perceber os padrões de movimento e de localização dos cidadãos para que os serviços de transporte possam ser dinamicamente ajustados às necessidades dos clientes. O desenvolvimento implica, por um lado, a optimização de uma aplicação/consola do lado do motorista do veículo relativamente a elementos técnicos como a forma de comunicação com outros dispositivos e o algoritmo de deteção de paragens, que vai acompanhar o veículo e permitir que a aplicação tenha noção do percurso e das paragens, com tudo o que isso implica no processo de vendas e carregamentos; por outro, a integração de cardlet NFC numa outra aplicação para o cliente, permitindo às pessoas usar a aplicação para comprar/carregar títulos de transporte e ter acesso aos horários dos serviços e outras informações, enquanto o uso desta aplicação por parte dos clientes terá influência nos mecanismos de recomendação de novas rotas e serviços, tanto regulares como *on-demand*.

A aplicação móvel para o motorista é uma das componentes incluídas num destes sistemas de gestão. Através da aplicação é possível que um motorista inicie o seu serviço - que pode ser regular ou não-

programado - e possa vender, carregar e validar títulos de transporte (bilhete, cartão contactless, NFC, QRCode...) com um determinado contrato/tarifa e com dependência da zona/localização.

Esta aplicação de consola do motorista comunica com o módulo de *back-office* do *ticketing* através de um conjunto de serviços SOAP. Estes serviços SOAP foram alterados para serviços REST através da utilização da API Retrofit, que é essencialmente um cliente REST para Android que permite implementar os serviços usando classes específicas para modelar a resposta recebida em JSON e uma interface de métodos com anotações que evidenciam o tipo do pedido. Esta conversão de serviços foi feita para melhorar a performance temporal e, consequentemente, a sincronização com a base de dados no *back-office*, já que é este o principal objetivo dos serviços web implementados.

A nível da comunicação da consola com os Terminais de Pagamento Automático (TPA), esta é feita através de sockets Bluetooth. Por questões de instabilidade observadas no terreno, a comunicação por sockets Bluetooth deve ser substituída por comunicação PCL (Payment Communication Layer), desenvolvida pela Ingenico para comunicação com os TPA Ingenico.

Para usar o PCL, há comandos específicos criados para cada ação que devem ser construídos em runtime. A grande maioria partilha a mesma estrutura base, mas pode haver alterações na mesma consoante o comando, no sentido em que mais informação pode ser adicionada se a ação em causa assim o exigir, nomeadamente transferência de ficheiros.

A implementação do PCL como forma de comunicação entre o dispositivo móvel e o terminal de pagamento implicou várias alterações e implementação de novos elementos que dependem da comunicação entre dispositivo e terminal e, portanto, passam a ter uma nova forma de operar.

Os dois modos de operação da aplicação, conhecidos como o modo de leitura e o modo de validação, foram parte do foco no que toca a estas alterações. A lógica de receção de cartões lidos ou informação de validações foi alterada para melhorar o desempenho da aplicação, resultando não só na reformulação da ordem, tipo e número de mensagens trocadas para este fim mas também na adição de novos elementos que terão de ser enviados aquando da ativação dos modos.

Dentro destes modos de operação determinadas funcionalidades precisaram de ser reformuladas para serem compatíveis com este novo protocolo de comunicação. A impressão de bilhetes/recibos foi uma destas funcionalidades, já que toda a lógica de impressão, incluindo o formato como os dados para imprimir são enviados para o terminal, foi alterada. Paralelamente, foi também importante ter em atenção casos de fraude que eram passíveis de acontecer durante o processo e tratá-los tendo em conta a lógica de negócio da empresa cliente.

A capacidade que a aplicação tem de conseguir enviar ficheiros completos para o terminal e usar alguns deles para atualizar determinados componentes de *software* do terminal foi também influenciada pela adição do PCL. A isto foi aliada a capacidade da aplicação do motorista conseguir fazer essas atualizações automaticamente, sem intervenção humana, estando dependente apenas de um sistema de difusão de ficheiros de forma remota que é completamente alheio aos outros módulos do sistema.

Também outras alterações foram feitas possibilitando o carregamento de título de transporte através de "listas brancas", a venda de bilhetes de papel fora do contexto do autocarro, criando um modo 'quiosque', e o envio para o *back-office* de informação mais completa sobre os turnos, serviços e vendas realizadas.

Outro módulo com que a aplicação comunica é o *back-office* de *tracking*. Este módulo é o responsável

pelos dados de localização que chegam à consola do motorista. Esses dados (coordenadas GPS) são tratados na componente Android de modo a que esta possa acompanhar automaticamente o percurso do autocarro e, consequentemente, automatizar determinados elementos da compra/venda de títulos de transporte, como as paragens/zonas do percurso. O algoritmo de cálculo necessário para tratar os dados de localização necessitou de optimizações para lidar, não só com casos práticos específicos capazes de quebrar o algoritmo, mas também para aumentar a precisão do mesmo.

No final, a implementação e/ou optimização destas soluções e desafios foi completada com sucesso, como demonstram os resultados que revelam uma maior estabilidade e melhor performance da aplicação. Com isto pretende-se aumentar a eficiência dos motores de operações comerciais e de gestão dos serviços de mobilidade, simplificando a interação entre motorista e passageiro, o que é particularmente importante num ambiente com intensa atividade turística. Estes aspectos aliados ao típico cenário de uma vida apressada requer uma plataforma que suporte o acompanhamento destes serviços de mobilidade face à evolução da sociedade e do modo de vida que lhe associado, evitando que haja um afastamento conceitual entre a necessidade e a oferta maior do que o que já existe hoje em dia.

Palavras-chave: Public transport; PCL; Mobility; Recommendation system; Android.

Abstract

Within the scope of the SMART CITY BUS: FLEXIBLE TRANSPORT.AI FRONT-END project, developed in the company Card4B - Systems, S.A. the present document was written to report the results and conclusions of said project in the context of the discipline of Computer Engineering Dissertation/Project Report of the 2nd year of MSc in Computer Engineering from the School of Sciences of the University of Lisbon. This project is directly related to public transportation services management and improvement regarding its flexibility and is focused in the front-end (Android) component of the system.

Nowadays, the public transportation services are almost exclusively divided in two main categories: regular collective services with predefined routes and schedule and on-demand services (such as UBER). Any additional flexibility in these services is reflected in its price, which inevitably alienates a big volume of the population. The main idea is to build a system to manage these services in a more dynamic way by getting information regarding people's movement patterns in the city. This system is to be integrated in a public transportation management platform.

The driver mobile application is part of one of those systems. This android component is supposed to be used by the bus drivers to open and close their service - which can be either a predefined service or not – and ultimately sell, charge and validate transport titles with a certain contract/tariff, depending on their current location.

The communication between the mobile app and the other elements of the system is indispensable to guarantee the reflection of the data collected by the services, shifts and sales performed in the application in the overall recommendation system. In the end, all the elements implemented and optimized along this project proved to be an asset for the entire system by improving the performance and stability of the Android front-end component.

Keywords: Public transport; PCL; Mobility; Recommendation system; Android.

Contents

Chapter 1

Introduction

The idea of a dynamic mobility management system that goes beyond the two main types - regular and on-demand - offered nowadays is not completely new. On the contrary, it is a need many public transportation companies have identified but not all of them have managed to develop a solution. Such a system can contribute to a bigger level of flexibility in these mobility services and allow it to be dynamically managed.

Artificial intelligence could be an asset in the creation of said system, using user input and observed behaviour as an opportunity to create services based on the client's needs. A mobile application, that should allow the customer to buy, charge and validate transport titles, could reveal information about its user and work as a intervening element in the recommendation system. Complementary to this, another application, now on the driver's side, can work as a selling and validation point, not only for typical transport titles, but also for the described user application.

1.1 Motivation

More then a step in the path of technological evolution in today's society, the need for a public transportation management platform that allows the creation of services suited for particular patterns of movement within a city is directly connected to the ever increasingly influential idea of individualism and expectation for easier alternatives. It is almost common sense that the majority of the population that uses public transport is quite displeased with the offered services. A study on the satisfaction levels of the people over the existent public mobility services [4] shows that the bus is the second most used and second least satisfactory transport mainly due to the scheduling, delays and vehicle coordination. Not only the local population requires a more efficient way to go from their home to their place of work everyday, but also the explosive numbers of tourists demand a serious improvement on the management of these services, specially in key locations in the city.

Also, considering most people nowadays live through their smartphones, this dependency could be allied with this type of mobility management. A digital improvement of the system in place today to buy, charge or validate transport titles would simplify the interaction between driver and client which would be specially useful in a touristic context. The entire process could be simplified, which would lead to an increase in the use of these services. Additionally, digital tickets are not as likely to be lost as physical tickets and are considered an eco-friendly alternative.

Furthermore, the need from the population for greater flexibility in mobility services is growing

due to today's fast paced life and ever increasing focus in culture and tourism and requires an efficient management system that will be able to support it and simplify its use by the masses, as well as modernize it.

In regards to the current state of the system, more specifically the front-end component, which is the primary focus of this project, a mobile application assists bus drivers in selling paper ticket, validating transport title cards and reading them, mainly. The application also supports GPS navigation, not only with an informative purpose, but also to allow the selling of tickets whose prices depend on location parameters.

Meanwhile, the driver app presents some challenges, mainly on the communication front: not only between application and back-office, where data replication takes too long, delaying the use of the application, but also between the application the Automatic Payment Terminal where the connection needed for the proper functioning of the application is highly unstable. Beyond that, the current operation logic, from the dispatch of files to the terminal to the message exchange logic in validation and reading environments, does not result in the best performance and needs to be optimized. Furthermore, the bus stop detection algorithm can not cope with certain practical cases that don't obey to the idea that a route is always done precisely the way it is supposed to.

1.2 Objectives

The main goals of the project are focused on the front-end component of the system, the driver application, and represent essentially the optimization of existing elements and the implementation of new ones.

- Optimization of the communication between the driver mobile application and the ticketing back-office component to improve the performance of the application;
- Optimization of bus stop detection algorithm used by the driver app in order to prevent cases where bus stops are wrongly detected in some practical cases;
- Implementation of a new form of communication between the driver application and the Automatic Payment Terminal (APT). Consequently, implement further elements based on this new form of communication.

1.3 Planning

The initial planning for the project had a start date of October 1st 2018 and an end date set for July 1st 2019. The project was divided in 7 greater tasks, including the writing of both the preliminary report and the final report

1. Conversion of the SOAP web services of the ticketing and tracking back-office modules called by the driver application console to REST web services using the Retrofit API and including a review of REST advantages over SOAP in this practical case - **2 weeks**;

2. Elaboration of Proof-of-Concept and implementation of PCL communication between the driver application and the APT devices, replacing the existent Bluetooth socket communication. Evaluation of time/performance of the PCL implementation and study of other eventual alternatives such as Wi-Fi communication - **2 months**;

3. Writing of the preliminary report - **2 weeks**;

4. Optimization of the bus stop detection algorithm in the driver application taking to account information received from the tracking back-office module and GPS coordinates, while also evaluating the use of KML and eventual available Google services - **1,5 months**;

5. Integration with APIs and external services (Google services, www.weather.com, ...). Initially performing a study on the available APIs and Google Services to see what types of services are available and could eventually be fruitful when integrated with the driver app - **1,5 months**;

6. Integration of the mobile ticketing libraries (Cardlet NFC/Beacon BLE) in the MYINFO mobile application, used by the passengers to validate cards with the driver application, have access to information and schedules, etc. Initial familiarization with MYINFO while adapting a preexisting version of the application to serve the needs of a certain company and later perform a re-engineering and restructuring operation in the app to integrate the mobile ticketing features - **3 months**;

7. Writing of the final report - **1 month**.

At the end of this project and considering the tasks described above, it started on October 1st 2018 and an end date of September 30th 2019, although not all of the initially proposed tasks were accomplished, namely tasks 5 and 6. This is due to the initially unforeseen extension of task 2 which required further implemented elements such as the generation and transference of files in run-time, the revision of printing and sale logic, as well as other elements described in this document.

1.4 Contributions

As a result of the developed work in this project, a new and enhanced version of the system emerged, revealing several improvements around the front-office component and its direct collaborative system components. The revision, correction and implementation of the communication, operational logic and several other elements led to a better performance from the driver mobile application in real life situations. The app users observe a faster and far more robust application which directly reflects on the overall stability of this technological solution.

Not only did the new communication logic allow the application to validate and sell transport titles in less time, it also improved connection stability, which led to better usability. It also contributed to faster back-office data sending and receiving, which is specially helpful in cases where data must be generated and sent to the back-office component in real-time.

The implementation of white list card loads and the Cash operation mode expanded the functionality of the application beyond its original borders, allowing the driver application to be used in different environment from the one originally defined and permitting a certain form of transport title card loads.

Essentially, the improvements made in the driver app are a reflection of today's client requirements for the public transportation industry, where every use case must be as easy, immediate and invisible as possible.

1.5 Institutional Framework

The project described in this document is inserted in the context of the discipline of Computer Engineering Dissertation/Project Report of the 2nd year of MSc in Computer Engineering from the School of Sciences of the University of Lisbon. The project was developed in Card4B - Systems, a company whose business focus is producing front-end and back-end ticketing software for public transportation companies.

1.6 Document Structure

- Chapter 2 - Overview of work related to the technological components used in this project and/or of similar goals as an overall project.
- Chapter 3 - Initial state of the driver mobile application and description of its surrounding system, including the different modules of the platform, their purposes and the description of all initial application functionalities and interface.
- Chapter 4 - Description of all the work accomplished during this project including all the newly implemented elements, performed corrections and alterations to the previous state of the driver application.
- Chapter 5 - Results of the performed implementations/alterations including statistic and comparative data to evaluate the improvements.
- Chapter 6 - Conclusion of the overall project and description of future work based on what has been accomplished so far.

Chapter 2

Related Work

In this chapter, a look is taken at some technologies that were used along this project as well as other systems revolving around public transportation services. It also includes a small study on state of the art for certain elements such as the SOAP and REST web services.

2.1 Web Services: SOAP Protocol vs. REST Protocol

Following the definition given by the World Wide Web Consortium, a web service is *a software application identified by a URI whose interface and bindings are capable of being defined, described and discovered as XML artifacts* [2]. These services could be used obeying to either one of the two main principles used: SOAP (SOAP based services) or REST (RESTful services).

The SOAP based services use XML for the message exchange in the service. This XML document is essentially a message containing an envelope, header and body which will later be sent with the help of a transport protocol. The RESTful services, on the other hand, do not require a message structure with envelope and header, therefore the XML parsing is avoided. The fact that REST is stateless and every transaction in unrelated to each other allied with the revoked need for the parsing makes REST simpler and lightweight with less bandwidth requirement, in comparison with SOAP. Due to the structural difference between the messages in both protocols, the message size is typically smaller in RESTful services rather than SOAP based ones, which significantly decreases the request/response time [3].

This differences can be a relevant argument in regards to the conversion of SOAP services into RESTful in many computational scenarios as technological evolution calls for simpler and faster computing elements.

2.2 PCL - Payment Communication Layer

The Payment Communication Layer (PCL) is a set of tools, libraries and documentation developed by Ingenico [1] used to setup and manage the communication component between an Android device and an Ingenico APT device. PCL serves as an abstraction point for the implementation of the Android-APT communication, since it is done through TCP sockets, while PCL handles the TCP <=> Bluetooth conversion in lower levels of the PCL software.

[1] https://www.ingenico.com

PCL supports several types of connectivity, namely Bluetooth, USB, IP and RS232. To connect the Android device to the terminal via Bluetooth, the devices must be paired. After being paired, it is possible to use the libraries of PCL to retrieve all paired devices and ultimately activate the one we wish to establish the connection with.

After the connection and activation of companion are complete, the PCL service must be bound and initiated after a designated dynamic bridge is set. Consequently, it allows to communicate with the device through a TCP socket.

2.3 Solutions with NFC

NFC is a wireless short-range technology that allows two devices separated by a short distance to communicate, it is compatible with contactless cards and is currently used in the ticket validation process in public transportation services. With the evolution of technology, the ever growing presence of NFC in devices such as smartphones is undeniable. Because smartphones are such a personal item and now contain said NFC technology, these devices end up being the perfect host for an electronic ticketing system integration.

A device with NFC can act in three different modes [1] :

1. Peer-to-peer - to establish a two-way communication between two devices;

2. Card emulation - to emulate a contactless smart card with a secure element (a component the allows the private information of the user to be stored);

3. Reader/writer - to allow the mobile device to read or write passive tags.

The card emulation operation mode is associated with the use of a secure element (SE) and therefore needs to have access to the correct keys, which may be controlled by different entities, depending on the case. The peer-to-peer mode, on the other hand, exchanges messages relying on certain protocols such as SNEP (Simple NDEF Exchange Protocol) - allows the exchange of NDEF messages in-between devices - and LLPC - allows to establish either connectionless or connection-oriented services. In the case of the reader/writer mode, although is it supposed to be used to read or write passive tags, it can also be used to exhange messages, as long as the communication pattern is known in advance.

Nowadays, already existent projects such as QuickTap [2] in the UK, PingPing [3] in Belgium, GoogleWallet [4] in the USA and PayPalNFC [5] use the NFC peer-to-peer operation mode and allow users to use their personal smartphone to make cashless purchases, essentially acting as a digital wallet.

In regards to electronic ticketing, particularly used for public transportation services, in Porto there is already an electronic ticketing solution associated with Andante [6], Porto's main transport title. The ANDA application works as a normal transport title card and consequently erases the need for a physical card. From the moment it is validated, it tracks the route of the customer and when the app detects the

[2] https://www.quicktap.co.nz
[3] https://www.pingping.be/nl/
[4] https://pay.google.com/
[5] https://www.paypal.com/
[6] https://www.linhandante.com

the passenger has left the bus, an automatically calculated price is presented to the user. This application requires and takes advantage of technologies such as Bluetooth, NFC, internet access and GPS location.

2.4 Summary

Overall, the SOAP/REST analysis is quite relevant considering one of the goals of the project is to evaluate the SOAP web services performance and convert it into REST if such change is justified by further tests and analysis. On the other hand, the Payment Communication Layer (PCL) is to be fully implemented as the new layer of communication between the Android device and the Automatic Payment Terminal. Furthermore, the mobile application, which is the focus of this project, will need to be able to use the NFC technology to validate transport titles through the bus passenger's smartphone, although this is something set as future work in the application.

Chapter 3

Analysis

This project revolves around an already existing system that needs further developments to evolve in regards to the goals and motivation of this project. This chapter describes the initial state of the system, detailing all the modules that integrate it while paying special attention to the driver mobile application, since it is the main focus for this project. All the features and functionalities of the application are described as well as some more technical elements whose information that may be relevant for the further development of this document.

Chapter 4

Implementation

Along the development of this project, the system was improved considerably after several changed and newly implemented elements. This chapter describes all of that developed work in detail while also considering the main goals for this project, from the web services that connect the driver application and the ticketing back-office to the bus stop algorithm optimization and the implementation of a new form of communication between the app and the APT, as well as all the work that resulted of this implementation.

Chapter 5

Results

Considering the work accomplished, several tests and measurements were performed to evaluate the improvements. This chapter's focus is a presentation of collected data in regards to the accomplished work described along this document, the interpretation of such data and eventually the comparison between the performance of the new state of the system and the performance of the system previous to these developments, proving the advantages of the newly implemented or revised components.

Chapter 6

Conclusions and Future Work

To conclude, a reflection on what was done allows us to understand the relevance and magnitude of the accomplished work. This chapter presents that reflection, including the main differences, additions and improvements, as well as the work that is yet to be done on the system, or more specifically in the driver application.

6.1 Conclusions

At the end of this project, the goal of improving the performance, usability and functionality of the driver mobile application was accomplished.

The replacement of SOAP web services for REST web services resulted in significantly better time performance in regards to data replication, while the complete change communication between the application and the connected automatic payment terminal also proved to be quite fruitful, resulting, not only time-wise, specially during card detection operations such as reading and validating transport titles, but also in regards to message exchange logic which was improved to the point of reducing the number of exchanged messages and, consequentially, reducing the probability of connection failure amidst a service or operation. The implementation of PCL also brought an enhancement concerning the previously verified random connection failures in real-life practical situations that caused a disturbance of the good functioning of the application.

Further implementations significantly improved several functionality points of the application, namely the automatic updates of different components of the system without significant human input, the generation of files required for navigation in run-time, avoiding large and timely downloads of files containing unnecessary information and the restructuring of the printing and sale logic to deal with possible fraud cases that could easily be replicated by the users.

All of these implemented and improved elements resulted in an application with greater levels stability, functionality and overall performance, creating a solid base for the growing of a product that will certainly be the solutions for future needs of the client companies and today's public transportation industry.

6.2 Future Work

Despite the work accomplished so far, several other elements remain to be implemented. The automatic download logic remains to be extended to different files such as external libraries the system will require in the near future, since it will start using such libraries and eventual updates for these files will need to be performed without human input, expect for the dissemination of the source file, as currently happens. Besides these external libraries, and similarly to the ALC file (also referred to as whitelist), a blacklist will also be created to register transport title cards that, for some reason, are not allowed to validate. This new blacklist file will possibly follow the logic of the ALC file and will be generated in run-time with data from the driver app database and downloaded into the APT.

Cash mode is also expected to sell card products as a card load. This operation will require the addition of a new component of the system which establishes a mean of communication between the application and the TK component, without using the APT. The TK component will be in charge of the charge operation itself, although the driver app will be used to choose products and required parameters. The application will also be responsible for the correct registration of the operation considering the result of the card load performed in the APT.

A new type of CardInteraction will also be implemented. These interaction will no longer represent a card but a ticket that was bought on the back-office component of the system. This sale will generate a QRCode for the client and a CardInteraction representing this sale. When the passenger uses the QRCode on the driver app, the application will need to register the sale and set its resulting external transaction in the corresponding CardInteraction.

Finally, a new operation mode will be introduced: Inspector mode. This mode will, not only present the card content on the screen of the application but also verify the validity of the transport title.

Bibliography

[1] U Biader Ceipidor, CM Medaglia, A Marino, M Morena, S Sposato, A Moroni, P Di Rollo, and M La Morgia. Mobile ticketing with nfc management for transport companies. problems and solutions. In *2013 5th International Workshop on Near Field Communication (NFC)*, pages 1–6, Computer Science Department - Sapienza University of Rome, Italy, February 2013. IEEE.

[2] Harumi Kuno Vijay Machiraju Gustavo Alonso, Fabio Casati. In *Web Services. Data-Centric Systems and Applications*, Germany, 2004. Springer, Berlin, Heidelberg.

[3] Snehal Mumbaikar and Puja Padiya. Web services based on soap and rest principles. In *International Journal of Scientific and Research Publications, Volume 3, Issue 5*, Mumbai, India, May 2013.

[4] Deco Proteste. Transportes públicos - metro é o que mais agrada. Portugal, February 2015.